Introducing NULL

In the previous lecture, you learned about the special NULL value - which simply means that a record has no data for a specific column.

As mentioned, NULL is not the same as 0 or 0.0 for numeric columns or '' (i.e. an empty string, empty text) for text columns. The latter examples (0, 0.0, '') are examples for values that are available - there is no missing data here 0 is a valid number after all.

How can you insert NULL as a value?

You simply insert it by omitting a column (and its value) in the INSERT INTO statement.

Here's an example:

Assume that you have a table that was created like this:

```
CREATE TABLE example (
  full_name VARCHAR(200),
  salary INT
);
```

Data could be inserted like this:

```
INSERT INTO example (full_name, salary)
VALUES ('Max Schwarz', 19000);
```

In this case, no NULL data would end up in the inserted row as values are provided for all columns.

Here's the same example without a salary:

```
INSERT INTO example (full_name)
VALUES ('Max Schwarz');
```

Only the full_name column is referenced (and receives a value). salary is omitted - both in the column as well as in the VALUES list. In this case, no data would be written into the salary column for this inserted row - i.e. NULL would be the assumed value for salary for this record.

If you would perform an operation like the below one, the NULL value would be ignored and would therefore not distort the overall result:

```
SELECT AVG(salary) FROM example; -- The average would be calculated without taking the NULL values into account (i.e. NULL is NOT treated as 0!)
```

NULL and DEFAULT Values

If you have a table definition where some (or all) columns have default values (defined via DEFAULT), those columns can never store NULL as a value - even if no specific data is provided in the INSERT INTO statement. The reason for that is, that the DEFAULT value will be used if no specific data is provided.